Link-uri utile:

https://rollbar.com/blog/java-exceptions-hierarchy-explained/ Aici gasiti ierarhia un pic mai extinsa. Pentru usurinta sa cautati mai apoi clasele in documentatia oficiala de la oracle.

https://www.w3schools.com/java/java\_try\_catch.asp

https://www.tutorialspoint.com/java/java\_exceptions.htm

https://stackoverflow.com/questions/2190161/difference-between-java-lang-runtimeexception-and-java-lang-exception

Problema 1.

Sa se creeze un program cu o clasa Main in care se va crea o metoda care citeste cuvinte de la tastatura pana la introducerea unui anume cuvant.

Daca cuvantul respectiv contine cuvinte precum ("pruna" sau "pupaza") se va arunca o exceptie. (Exceptia trebuie creata de voi).

(Mai pe scurt simulam un mic programel in care prevenim cuvintele vulgare). :)

Problema 2.

Sa se creeze un program simplu in care se va crea o metoda (in acceeasi clasa cu metoda main).

Metoda va primi ca parametru un array de numere (int).

Daca numerele din array nu sunt in ordine consecutiva se va arunca o exceptie.

Exemplu:

{5,6,7,8} - nu va arunca exceptie.

{5,6,9,8} - va arunca exceptie.

Problema 3.

Sa se creeze un program prin in care se creeaza o clasa Servicii ce va desemna anumite functii pentru gestiunea unor Persoane inr-un sistem.

Se vor trata cazurile de nefunctionare corespunzatoare prin exceptii. Pentru aceasta se va crea o clasa ServiceException (trebuie sa fie o exceptie. Urmati exemplul de la curs).

Cerinte clasa Person

1. Pentru aceasta se va creea clasa Persoana cu fieldurile: numeComplet, CNP (pentru usurinta luati-l de tip long)

2. Se vor genera metodele equals si hashcode.

3. Se va genera metoda toString(). (Pentru a afisa detaliile persoanei. nu trebuie apelat super.toString() cum am mai vazut prin teme).

Cerinte clasa Servicii.

1. Clasa servicii va avea o lista de Persoane.

2. Se va implementa o metoda care va primi ca parametrul un numar de tip long ce va reprezenta CNP-ul unei persoane.

2.1 Metoda va verifica daca daca exista o persoana cu acest CNP in lista.

2.1.1. Daca CNP-ul exista in lista se va afisa numele persoanei

2.1.2. Daca CNP-ul nu exista in lista se va arunca o exceptie.

2.2 Metoda care introduce o persoana in lista. Metoda primeste ca parametru un obiect de tipul Person

2.2.1 Daca persoana exista deja in lista se va arunca o exceptie.

2.2.2. (SUPLIMENTAR) Daca exista o alta persoana cu acelasi CNP dar nume diferit se va arunca o exceptie. (Nu exista 2 persoane cu acelasi CNP in realitate).

EXEMPLE:

1.ARRAYS

|  |
| --- |
| package com.itfactory.arrays; |
|  |  |
|  | public class Arrays { |
|  | public static void main(String[] args) throws Exception { |
|  | int[] array = {1, 2, 3, 4}; |
|  | try { |
|  | for (int i = 0; i < 5; i++) { |
|  | System.out.println(array[i]); |
|  | } |
|  | } catch (ArrayIndexOutOfBoundsException e) { |
|  | throw new Exception("S-a mers prea departe"); |
|  | } finally { |
|  | System.out.println("Alt mesaj"); |
|  | } |
|  | } |
|  | } |

2. DivideBy0Exception:

|  |
| --- |
| package com.itfactory.divizare; |
|  |  |
|  | public class DivideBy0Exception extends Exception { |
|  |  |
|  | public DivideBy0Exception(String message) { |
|  | super(message); |
|  | } |
|  | } |

3.DIVIZARE

|  |
| --- |
| package com.itfactory.divizare; |
|  |  |
|  | import java.util.Random; |
|  |  |
|  | public class Divizare { |
|  |  |
|  | public static void main(String[] args) { |
|  |  |
|  | try { |
|  | System.out.println(impartireSafe(10, 2)); |
|  | System.out.println(impartireSafe(10, 0)); |
|  | System.out.println("Am trecut fara nicio eroare"); |
|  | } catch (DivideBy0Exception e) { |
|  | e.printStackTrace(); |
|  | } |
|  | } |
|  |  |
|  | public static int impartireSafe(int numar1, int numar2) throws DivideBy0Exception { |
|  | System.out.println("Execut impartirea"); |
|  | if (numar2 == 0) { |
|  | throw new DivideBy0Exception("S-a incercat divizarea la 0"); |
|  | } |
|  | return numar1 / numar2; |
|  | } |
|  | } |

Give feedback

4. ERROR

|  |
| --- |
| package com.itfactory.error; |
|  |  |
|  | public class OutOfMemory { |
|  |  |
|  | public static void main(String[] args) { |
|  | String[] array = {}; |
|  | main(array); |
|  | } |
|  |  |
|  | } |

5. EXCEPTII

|  |
| --- |
| package com.itfactory.exceptii; |
|  |  |
|  | public class ExceptiaMea extends Exception { |
|  |  |
|  | public ExceptiaMea(String mesaj) { |
|  | super(mesaj); |
|  | } |
|  |  |
|  | } |

6. STOCURI\_EXCEPTII

|  |
| --- |
| package com.itfactory.stocuri.exceptions; |
|  |  |
|  | public class InvalidStockException extends Exception { |
|  | public InvalidStockException(String mesaj) { |
|  | super(mesaj); |
|  | } |
|  | } |

7. EXCEPTII\_STOCURI\_SERVICII\_PRODUS

|  |
| --- |
| package com.itfactory.stocuri.servicii; |
|  |  |
|  | public class Produs { |
|  | private int serie; |
|  | private String categorie; |
|  | private double pret; |
|  |  |
|  | public Produs(int serie, String categorie, double pret) { |
|  | this.serie = serie; |
|  | this.categorie = categorie; |
|  | this.pret = pret; |
|  | } |
|  |  |
|  | @Override |
|  | public String toString() { |
|  | return "Produs{" + |
|  | "serie=" + serie + |
|  | ", categorie='" + categorie + '\'' + |
|  | ", pret=" + pret + |
|  | '}'; |
|  | } |
|  | } |

8.EXCEPTII\_STOCURI\_SERVICII\_STOCURI

package com.itfactory.stocuri.servicii;

import com.itfactory.stocuri.exceptions.InvalidStockException;

import java.util.ArrayList;

import java.util.List;

import java.util.Map;

/\*\*

\* Categorii: imbracaminte, electronice

\*/

public class Stocuri {

private List<Produs> stockProduse;

public Stocuri() {

stockProduse = new ArrayList<>();

}

public void genereazaStockDeProduse(Map<Integer, String> categorii) throws InvalidStockException {

for (Map.Entry<Integer, String> entry : categorii.entrySet()) {

//Extrag cate o pereche din map

int serie = entry.getKey();

String categorie = entry.getValue();

//Verific categoria: Trebuie sa fie doar imbracaminte sau electronice

if (!(categorie.equals("imbracaminte") || categorie.equals("electronice")) ) {

throw new InvalidStockException("Stoc invalid");

}

stockProduse.add(new Produs(serie, categorie, 123.5));

}

}

public List<Produs> getStockProduse() {

return stockProduse;

}

}

9. EXCEPTII\_STOCURI\_SERVICII\_PRODUS\_STOCURI\_MAIN

|  |
| --- |
| package com.itfactory.stocuri; |
|  |  |
|  | import com.itfactory.stocuri.exceptions.InvalidStockException; |
|  | import com.itfactory.stocuri.servicii.Stocuri; |
|  |  |
|  | import java.util.Map; |
|  |  |
|  | public class Main { |
|  | public static void main(String[] args) { |
|  | Stocuri stocuri = new Stocuri(); |
|  | try { |
|  | Map<Integer, String> stock = Map.of(101, "electronice", 102, "electronice", 103, "imbracaminte"); |
|  | stocuri.genereazaStockDeProduse(stock); |
|  | } catch (InvalidStockException e) { |
|  | e.printStackTrace(); |
|  | } |
|  | System.out.println(stocuri.getStockProduse()); |
|  | } |
|  | } |

10. TIPURI DE EXCEPTII\_CHECKED\_CITIRE FISIER

|  |
| --- |
| package com.itfactory.tipurideexceptii.checked; |
|  |  |
|  | import java.io.IOException; |
|  | import java.nio.file.Files; |
|  | import java.nio.file.Path; |
|  | import java.nio.file.Paths; |
|  | import java.util.List; |
|  |  |
|  | public class CitireFisier { |
|  | public static void main(String[] args) { |
|  | Path path = Paths.get("fileTest.txt"); |
|  |  |
|  | List<String> read = null; |
|  | try { |
|  | read = Files.readAllLines(path); |
|  | System.out.println("Alt mesaj"); |
|  | System.out.println(read); |
|  | System.out.println("Mesaj"); |
|  | } catch (IOException exception) { |
|  | System.out.println("Fisierul nu este prezent"); |
|  | exception.printStackTrace(); |
|  | } |
|  | } |
|  | } |

11. TIPURI DE EXCEPTII\_UNCHECKED\_CITIRE NUMERE

|  |
| --- |
| package com.itfactory.tipurideexceptii.unchecked; |
|  |  |
|  | import java.util.ArrayList; |
|  | import java.util.List; |
|  |  |
|  | public class CitireNumere { |
|  | public static void main(String[] args) { |
|  | List<String> numereStrings = new ArrayList<>(); |
|  | numereStrings.add("1"); |
|  | numereStrings.add("3"); |
|  | numereStrings.add("6"); |
|  | numereStrings.add("1a"); |
|  |  |
|  | List<Integer> numereTransformate = new ArrayList<>(); |
|  | for (String numereString : numereStrings) { |
|  | numereTransformate.add(Integer.parseInt(numereString)); |
|  | } |
|  | System.out.println(numereTransformate); |
|  | } |
|  | } |

12. TIPURI DE EXCEPTII\_UNCHECKED\_DIVIZAREA CU 0

|  |
| --- |
| package com.itfactory.tipurideexceptii.unchecked; |
|  |  |
|  | import java.util.Random; |
|  |  |
|  | public class DivizareaCu0 { |
|  | public static void main(String[] args) { |
|  | Random random = new Random(); |
|  |  |
|  | int random1 = random.nextInt(5); |
|  | int random2 = random.nextInt(2); |
|  |  |
|  | if (random2 != 0) { |
|  | System.out.println(random1 / random2); |
|  | } |
|  | } |
|  | } |

13. TIPURI DE EXCEPTII\_UNCHECKED\_NULL POINTER

|  |
| --- |
| package com.itfactory.tipurideexceptii.unchecked; |
|  |  |
|  | import java.util.ArrayList; |
|  | import java.util.List; |
|  |  |
|  | public class NullPointerExample { |
|  | public static void main(String[] args) { |
|  | List<String> produse = new ArrayList<>(); |
|  | produse.add("carte"); |
|  | produse.add(null); |
|  | produse.add("Haine"); |
|  |  |
|  | for (String s : produse) { |
|  | if (s != null && s.equals("Haine")) { |
|  | System.out.println("Am haine"); |
|  | } |
|  | } |
|  | } |
|  | } |

14.MAIN

|  |
| --- |
| package com.itfactory; |
|  |  |
|  | import com.itfactory.exceptii.ExceptiaMea; |
|  |  |
|  | import java.util.Scanner; |
|  |  |
|  | public class Main { |
|  |  |
|  | public static void main(String[] args) { |
|  | System.out.print("Citeste cuvant: "); |
|  | try { |
|  | String text = citesteCuvant(); |
|  | System.out.println(text); |
|  | } catch (ExceptiaMea exceptiaMea) { |
|  | exceptiaMea.printStackTrace(); |
|  | } |
|  |  |
|  | } |
|  |  |
|  | public static String citesteCuvant() throws ExceptiaMea { |
|  | Scanner scanner = new Scanner(System.in); |
|  | String mesaj = scanner.nextLine(); |
|  | if (mesaj.isEmpty()) { |
|  | throw new ExceptiaMea("Ati introdus un string gol"); |
|  | } |
|  | System.out.println("Sunt in metoda citeste cuvant"); |
|  | return mesaj; |
|  | } |
|  | } |